UDP/IP Integration Software Testbench Instructions

Team Gamma – Patrick Gauvin

April 17, 2017

Dependencies

- GCC
- make
- cmp (for the `check` make target)
- Python 2 (optional, required for test generators)
- Scapy for Python 2 (optional, required for test generators)
- `udp-executable-spec.zip`

Procedure

1. Unzip `udp-executable-spec.zip`.
2. Invoke `make` in the extracted folder to build the executable spec.
3. Invoke `make check` to run the executable spec with stored tests, the following is the expected output:

```
rx-odd pass
rx-odd2 pass
rx-even pass
rx-zero-len pass
tx-odd pass
tx-odd2 pass
tx-even pass
tx-zero-len pass
```

Example Usage

The `udp` program's usage message is the following:

```
Usage:
        ./udp <rx|tx> [--verbose|-v]

        Input is read from stdin, output is sent to stdout. In verbose
        mode, extra information about the transaction is printed to stderr
```

- An example invocation emulating the UDP receiver:

```
$ ./udp rx --verbose  < tests/rx-even.bin | hexdump -C
Source Address: 127.0.0.1
Destination Address: 1.2.3.4
Source Port: 60001
Destination Port: 60000
UDP Header Checksum: 0x3fa7
Data Length from Header: 0x2
Data Length from Datapath: 0x2
Error: 0
00000000  7f 00 00 01 ea 61 ea 60  68 69                    |.....a.`hi|
0000000a
```

- An example invocation emulating the UDP transmitter:

```
$ ./udp tx --verbose < tests/tx-even.bin | hexdump -C
Source port: 60001
Destination port: 60000
Length: 10
Checksum: 0x3fa7
00000000  7f 00 00 01 01 02 03 04  11 ea 61 ea 60 00 0a 3f  |..........a.`..?|
00000010  a7 68 69                                          |.hi|
00000013
```

Test Data Generators

These test generators create the binary input files which the udp program takes as input.

udp_rx_in_gen.py

```
$ python2 udp_rx_in_gen.py --help
usage: udp_rx_in_gen.py [-h] [--data DATA] [-o FNAME_OUTPUT] [-i FNAME_INPUT]
                        src dst sport dport

Create input-data files for UDP RX

positional arguments:
  src              IPv4 source address, e.g., 127.0.0.1
  dst              IPv4 destination address, e.g., 127.0.0.1
  sport            UDP source port
  dport            UDP destination port

optional arguments:
  -h, --help       show this help message and exit
  --data DATA      Data for the UDP payload (string)
  -o FNAME_OUTPUT  Output file
  -i FNAME_INPUT   Data input for the UDP payload, overrides --data
```

udp_tx_in_gen.py

```
$ python2 udp_tx_in_gen.py --help
usage: udp_tx_in_gen.py [-h] [-o FNAME] src dst sport dport data

Create input-data files for UDP TX
```

```
positional arguments:
  src         IPv4 source address, e.g., 127.0.0.1
  dst         IPv4 destination address, e.g., 127.0.0.1
  sport       UDP source port
  dport       UDP destination port
  data        Data for the UDP payload (string)

optional arguments:
  -h, --help  show this help message and exit
  -o FNAME    Output file
```

A External Software Links

- Cygwin (can provide tools like GCC, make, and cmp)
- Python
- Scapy